

University of Groningen

Color Processing using Max-trees

Tushabe, Florence; Wilkinson, M.H.F.

Published in:
Systems and Informatics (ICSAI), 2012 International Conference on

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2012

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Tushabe, F., & Wilkinson, M. H. F. (2012). Color Processing using Max-trees: A Comparison on Image Compression. In *Systems and Informatics (ICSAI), 2012 International Conference on* (pp. 1374-1380). IEEE (The Institute of Electrical and Electronics Engineers).

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Color Processing using Max-trees: A Comparison on Image Compression

Florence Tushabe
School of Computing and IT
Makerere University, Uganda
tushabe@cit.mak.ac.ug

M. H. F. Wilkinson,
Institute for Mathematics and Computing Science
University of Groningen
The Netherlands
m.h.f.wilkinson@gmail.com

Abstract—This paper proposes a new method of processing color images using mathematical morphology techniques. It adapts the Max-tree image representation to accommodate color and other vectorial images. The proposed method introduces three new ways of transforming the color image into a gray scale image that is filtered using conventional methods. Three new color reconstruction mechanisms are also proposed. The best method improves color fidelity by as much as 15%. The performance of six attribute filters are also compared on a jpeg compression operation.

I. INTRODUCTION

Connected filtering is a branch of mathematical morphology that filters connected components (connected sets of pixels of maximal extent) instead of individual pixels [1], [2], [3], [4]. A component is either retained as it is, or is removed if it does not satisfy given conditions, or *attribute criterion* [5]. Connected filters are therefore shape preserving and do not cause blurring even at high filtering levels. They allow users to chose properties of sections of the image that can be ignored, over-processed or filtered out. Connected filters have been used in various application, including image filtering and noise reduction [6], [7], [8], image simplification for compression [9], [10], video processing [10], [2], vessel enhancement filtering [11], [12], [13], and image analyzed microscopy [14]. Recent reviews can be found in [15], [16].

An important class of connected filters is based on the Max-tree and its dual the Min-tree [2]. Both are also referred to as component trees [17], [18], [8]. The aim of these trees is to encode a hierarchy of connected components at different levels to allow fast filtering or for analysis as a multi-scale representation of the image or volume under study.

In the framework of mathematical morphology, the basic working structure is a complete lattice [19], [20]. A complete lattice is a set of ordered elements (either partial or total order) for which each family of elements possesses a supremum (sup) or an infimum (inf) [19], [20]. Examples in image analysis are the lattice of subsets of the image domain in the case of binary images, and the lattice of scalar functions on the image domain in the case of gray scale images. The choice of suprema and infima in gray scale morphology is straight forward because gray level intensity values are completely ordered from black to white. In general, so long as the pixel values have a total order, whether range images, intensities,

or saturations, choosing the supremum (infimum) over the lattice of images consists of choosing uniform images filled with the supremum (or infimum) pixel value. By contrast, vector images, or images of scalars lacking total order such as hue or orientation, are not easily given a partial order. This poses problems in component-tree-based processing, because the hierarchy in the tree is driven by the total order of the pixel values. To build such trees in color morphology, the ordering has to be decided upon [21], [6], [8].

There are several types of multidimensional vector orderings [22], [21], [6]. A marginal ordering deals with each component independently and then later concatenates the scalars back together. This method has been shown to introduce new colors [8]. Reduced ordering obtains a scalar value from the vectorial components. Most researchers who adopt this approach calculate distances from a reference vector. In [23] colors are ordered with respect to their distances from white or black. In [6] color is ordered based on its distance from other reference colors which are not necessarily white or black. In [21] the minimum spanning tree of a region adjacency graph (RAG) is used.

Multivariate processing in color is generally approached in two major ways [24]. In marginal processing, each channel is processed independently, filtered using regular gray-scale morphology and then merged back into a single color image again. This approach has been found to be very efficient for denoising applications but poor at object detection [8]. The second approach is the vectorial one which transforms the multichannel data into a single channel based on one or more channels, processes it and then performs the color reconstruction.

Image filtering implemented using the Max-tree approach [2] is one of the fastest and most flexible ways of implementing connected filters [25]. Unfortunately, very little literature is available about how connected color processing is implemented by using the Max-tree approach. In [8], several orderings are investigated including marginal, lexicographic and reduced orderings. Four of the five tested approaches were found to produce undesirable colored artifacts and the one that did not suffered from very visible quantization effects.

This paper proposes a new Max-tree adaptation to color image processing that does not result in undesirable color artifacts or visible quantization effects. We propose a vecto-

rial image processing method in which the color vector is transformed into a scalar channel through a reduced ordering. Image reconstruction is carried out in such a way that it does not result in undesirable visible quantization effects or color artifacts.

The rest of the paper is organized as follows. In Section II, we discuss the theoretical aspects of the proposed method. Section III discusses the algorithms that were used in detail while Section IV presents the results obtained after the proposed method is tested on an image compression benchmarking dataset. Conclusions are then provided in Section V.

A. THEORY

Connected filters are image transformations that result in removal or retention of the connected components of an image. Practical implementations of connected filtering has been performed by using three major approaches. The pixel-queue algorithm [5], [17], [26], the Max-tree approach [2], [18] and the union-find method [25], [27]. This work deals exclusively with the Max-tree implementations.

B. The Max Tree

The Max-Tree [2] data structure is an efficient multi-scale representation of a grey scale image. The nodes C_h^k , with k the node index and h the gray level of the Max-Tree represent peak components P_h^k for all threshold levels in a data set. The root node represents the set of pixels belonging to the background, and each node has a pointer to its parent.

The filtering process is separated into three stages: construction, filtering and restitution. During the construction phase, the Max-tree is built from the flat zones of the image, collecting auxiliary data used for computing the node attributes at a later stage.

Once the attributes have been stored in the Max-Tree nodes, we can apply the attribute criterion of choice to each node to decide whether or not they should be retained. Various strategies of filtering are discussed in [2], [14], [28]. In all cases, filtering is performed by identifying and removing the nodes that do not fulfill the attribute criterion Λ .

The final phase is restitution, which consists of transforming the output Max-tree into an output image. If a node has been removed, a new gray level has to be assigned to it. Generally this is the gray level of the nearest preserved ancestor in the tree [2]. As a result, the gray level values of the original image are assigned to the pixels of the preserved nodes, and no new gray levels appear in the image.

If the criterion Λ is increasing, restitution is simple, because the tree is always pruned: i.e. if a node is rejected *all* its descendants are also rejected. However, if the criterion is not increasing, as in the case of scale-invariant filters [29], [14], there is a problem. Some rejected nodes have preserved descendants. There are several possible restitution decisions that can be made [5], [29], [14], [2]

Min: removes a node if any of its ancestors is removed.

Max: preserves a node if any ancestor is preserved.

Viterbi: treats selecting a correct pruning point in a branch of the tree as an optimization problem.

Direct: leaves all preserved nodes at their original gray value.

Subtractive: if a node is removed, all its descendants are lowered by the same amount.

The Max-tree is used for removing bright features. Removing dark features is done using a Min-tree, which is just the Max-tree of the inverted image.

C. Color connected filters

Color image processing differs little from gray scale processing. The image is first split into its component R, G, and B channels, then processed using the conventional means before the results are recombined into a color image again. This *marginal processing* is simple, and often effective, but can result in the appearance of new colors not previously present in the image. These “false colors” can present really nasty artefacts in images, and can be avoided by applying vectorial processing. Despite these objections, in noise removal using connected filters, marginal processing yielded the best results [8]. This is not surprising for two reasons: (i) because the filters are connected, no false edges can appear, as in normal morphological or linear filters [8], and (ii) noise is typically generated by independent processes in each R, G, and B channel. No correlations should exist. However, in HLS or $L^*a^*b^*$ spaces correlations in the noise do exist, and in that case vectorial processing should be better. In more general filtering tasks, even RGB representations should probably be treated in a vectorial way.

Several approaches have been proposed to solve these problem. The implementation issues involved in color connected operators can be divided into three parts: identification of the extremal points, the merging criteria of the removed region(s) and the color assignment to both the flat zones as well as the new merged region [30].

There are a few color connected filters that have been implemented. In the vector area morphology sieves approach (VAMS) [31], the supremum region is obtained by calculating aggregate distances between each flat zone and its connected neighbors. The extremal node is chosen as the one with the greatest aggregate distance. Merging is to the nearest neighbor node and the merged node takes on the color of the nearest neighbor while the flat zones adapt the mean color value in a given node. Another connected color filter is the convex color sieves (CCS) approach [32]. CCS is similar to VAMS except in the way that the extremal points are determined. In CCS, ordering is by first constructing a convex hull of each region and its connected neighbors. The extremal region is then defined as the one that lies on the edge of the hull.

It is interesting to note that VAMS [31] and CCS [32] process extremas without necessarily classifying them as either maxima or minima. This is because it is possible for that approach to obtain several connected extrema. This weakness has been dealt with by the introduction of the VAMOCs [7] which combines strengths of the VAMS and CCS methods,

and provides area openings and closings. Neither of these three methods explicitly builds a tree, because they use local order. This means that it is more difficult to perform fast multi-scale analysis as in [25], [14].

Another approach is that of the binary partition tree (BPT) [33], [34]. In this case the tree does not contain regional maxima (which by their nature require a total order or preorder) as their leaves, but the flat zones of the image, these are hierarchically merged using some measure of color *difference* to determine the merging order. Though highly effective in color image processing, their computation is not as fast as that of Max-trees, which explains the continuing interest in the latter.

One implementation of color connected filters that uses the Max-tree has been conducted by Naegel and Passat [8]. The Max-tree method requires that the data are ordered, which is easy in grey scale, but non-trivial in color space. Therefore, [8] impose either a total order or a total preorder on the color data. Let \mathcal{T} be our color space. A total order on \mathcal{T} is any binary relation \leq which is

- 1) reflexive: $a \leq a$ is true
- 2) transitive: $a \leq b \wedge b \leq c \Rightarrow a \leq c$
- 3) total: $(a \leq b) \vee (b \leq a)$ is true
- 4) antisymmetric: $(a \leq b) \wedge (b \leq a) \Rightarrow a = b$

In the case of a total *preorder* the last property (antisymmetry) does not hold. This means that if we use a total preorder on the color space to sort the pixels into different levels of the Max-tree, pixels with different colors can end up in the same node. A simple example would be sorting by the luminance of each pixel. Obviously, the first three relationships hold, due to the total ordering of luminance, but the last does not, because different color stimuli may have the same luminance. By contrast, the hue component from HSV or HLS color spaces cannot be used because it is not totally ordered.

In [8], the performance on noise removal of five different color connected area filters was tested, using four (pre)ordering schemes: (i) through marginal processing, (ii) using lexicographic ordering giving priority to R, G and B bands (in order of priority), (iii) a total order built by combining a total preorder based on the distance to the color white, and complementing it with lexicographic order, (iv) total preordering that calculates the distance of a node from color white (which is more or less equivalent to luminance).

Multiple color assignments within the restitution decisions were also tested. These apply to the preorder only, because in the case of total order we can simply use the existing rules for gray scale. The P_{mean} restitution rule assigns each node of the tree the mean value of its constituent pixels as its representative color, and then uses this representative value to reconstitute the nodes. This means that the rejected nodes obtain the representative color of the nearest preserved ancestor, *and* that the preserved nodes are assigned their own representative value.

The P_{median} decision is similar but uses the median color of the pixels in the region after sorting using lexicographic order. The results from [8] show that of all the methods that were

tested, only P_{mean} reconstruction did not introduce undesired colored artifacts. It however, altered the image quantization so much that it was very visible even at low thresholds. Indeed, even if the area threshold is set to zero, and all nodes are preserved, colors of pixels change. This is highly undesirable.

This work proposes different restitution decisions for color filtering using the Max-tree and compares the results with those after using the P_{mean} [8].

II. THE PROPOSED EXTENSIONS TO COLOR MAX-TREES

In this work we explore several different extensions to color Max-trees. Because marginal processing proved best in noise filtering, we focus on the application of image simplification for compression, extending the gray-scale work in [9]. First we explore different preorders in Section II-1, trying to find simple schemes with psycho-visually sensible orderings. We discuss extensions to the restitution decisions for color processing based on preorders, improving the results of [8]. Next, we explore different attributes suitable for simplification in Section II-2. Some of these are not new, but new algorithms for computing them needed to be developed.

1) *Orders and preorders*: The kind of ordering that is proposed in VAMS [31] and CCS [32] is not a total ordering because it is local. The same holds for the ordering proposed in VAMOCs [7]. The ordering in [8] is according to how far a color is from color white. This gives a higher priority to color white, thereby implying that white is more important than other colors. This poses the question of which color to use as reference.

We suggest that color is ordered according to the meaning behind it. All channels of a given color space represent a more generalized concept. For example saturation or chromaticity is represented in the S channel of the HLS and HSV color spaces and the second and third channels of the L*a*b* color space. Luminance is represented in the L channel of the HLS and L*a*b* color spaces, e.t.c.

Preordering color based on these channels gives the user better intuition of which channel would achieve the best result in this case. We propose a preordering based on:

- Chromaticity C ; defined as the length of the vector formed by the two chromaticity (color) components in the CIE L*a*b* color space [35]:

$$C = \sqrt{a^2 + b^2}. \quad (1)$$

with a and b the second and third component of the L*a*b* color space.

- Luminance (L_{Lab}); defined as the first component of the CIE L*a*b* color scheme [35].
- HLS Luminance (L_{HLS}); defined as the second component of the HLS color scheme [22]

$$L_{HLS} = 0.299R + 0.587G + 0.114B. \quad (2)$$

- Saturation (S); the third component in the HLS color space [22]

$$S = \begin{cases} 0 & \text{if } V = 0 \\ (V - X)/V & \text{otherwise} \end{cases} \quad (3)$$

with if $V = \max(R, G, B)$ and $X = \min(R, G, B)$.

- **Weighted Luminance (wL);** Luminance and chromaticity have been combined by giving luminance a higher weight according to Equation 4. In these experiments, $w_1 = 1$ and $w_2 = 256$.

$$wL = w_1C + w_2L_{Lab} \quad (4)$$

- **Weighted Chromaticity (wC);** Luminance and chromaticity are combined by giving chromaticity a higher weight according to Equation 5. In these experiments, $w_1 = 256$ and $w_2 = 1$.

$$wC = w_1C + w_2L_{Lab}. \quad (5)$$

2) *Attributes:* Filtering is achieved by determining whether an attribute value of each node satisfies a given attribute criterion. These experiments tested the following attributes:

- **Area:** The Area attribute calculates the size of a component and has been defined as [36], [26], [25]:

$$A(X) = \sum_{x \in X} \chi(X)(x) \quad (6)$$

where X is the set of pixels in the region, and $\chi(X)$ is the characteristic function of x .

- **Volume:** The Volume attribute [37] is the change in intensity over the area of a node and is given as:

$$V(X, f, h_{\text{parent}}) = \sum_{x \in X} (f(x) - h_{\text{parent}}) \quad (7)$$

where f is the intensity value within the original region and h_{parent} is the grey level of the parent.

- **Power:** The power attribute [10] calculates the square of the change in intensity over the area of a node. It is defined as:

$$P(X, f, h_{\text{parent}}) = \sum_{x \in X} (f(x) - h_{\text{parent}})^2 \quad (8)$$

- **Entropy:** The entropy attribute [38] measures the information content in the grey level distribution of a node and is defined as:

$$E(X) = - \sum p(f(x)) \log_2(p(f(x))), \quad (9)$$

with $p(f(x))$ the probability that $f(x)$ occurs within X .

- **Vision:** The Vision attribute [9] calculates the volume of all nodes but only components whose volume is equal to the threshold are considered.
- **VisionP:** We define the VisionP attribute to calculate the power of all nodes but only components whose power is equal to the threshold are considered.

A. The Proposed Method

In general, the proposed method uses two images: (i) the original color image (ORI), and (ii) the ORDER image that is generated from ORI and indicating the (pre)order of the pixels. ORDER image can represent either ORI's luminance, chromaticity or saturation.

1) *The Building Phase:* The Max tree is built from the ORDER image. Each component / node is assigned the mean color value of all pixels in that node. Attribute information is obtained from each node in the tree. In this work, we store the following attribute values: Entropy, Power, Vision, Area, Volume and VisionP. These are calculated as shown in Section II-2

2) *Filtering:* Once the tree has been constructed, the nodes that do not fulfil the user requirements are removed using the conventional gray-scale filtering rules (a pre-determined criteria).

3) *Restitution:* Image restitution is then performed by mapping to a given color from ORI image. There are many choices we can make concerning the final colors assigned to each pixel. This is because a single level in the ORDER image may correspond to multiple colors in ORI. Naegel and Passat [8] use the vector mean color or median color based on lexicographic ordering to represent preserved nodes, which leads to severe color artefacts even in preserved regions.

Here we propose a different approach: simply retain the original color of each pixel in each preserved node. This avoids all color artefacts and hence improves upon the work of [8]. Removed nodes must however be assigned a new color. For this we can copy the strategy of the P_{mean} decision for removed nodes, and assign the mean color of the closest surviving ancestor. This rule is referred to as the *Mean of Parent (MP)* decision, and only differs from P_{mean} in the treatment of preserved nodes.

Alternatively, we propose the *Nearest Color (NC)* approach. In this case a removed node selects the color closest to its own mean color from those adjacent pixels that belong to the nearest preserved ancestor. This guarantees that no new or false colors appear, because the color selected was always present in the image. We simultaneously minimize the color change while guaranteeing that the output image contains no unwanted structures.

The final rule is *Nearest Neighbor (NN)*. This assigns each pixel in the node to be filtered the color of the *spatially* nearest pixel in the first preserved ancestor. Unlike the other restitution rules, this splits up the removed nodes into different zones. This means it is no longer a connected filter in the classical sense, but it does prevent false colors and minimizes the edge strength along the boundary of the removed region. In a way, it could be seen as a quick-and-dirty version of the image inpainting as proposed in [39], which also aims at reducing the boundary between removed and preserved regions. Unlike [39], our method guarantees idempotence, because in the according to the preorder the new region is completely flat. This is because all colors used come from a single node in the Max-tree.

III. EXPERIMENTAL RESULTS

The experiments used the fourteen (14) test images obtained from the image compression benchmark database in [40]. Quality is measured using the mean Structural SIMilarity (SSIM) quality index [41]. The overall SSIM value of a given

TABLE I

QUALITY OBTAINED AFTER FILTERING USING THE DIFFERENT DECISIONS

Order	NN	NC	MP	P_{mean}	AvQuality
L_{Lab}	0.80	0.80	0.79	0.76	0.786
C	0.81	0.82	0.72	0.48	0.705
wL	0.79	0.79	0.79	0.79	0.789
wC	0.75	0.75	0.75	0.74	0.749
L_{HLS}	0.79	0.79	0.78	0.76	0.781
S	0.71	0.71	0.62	0.52	0.642
AvQuality	0.775	0.777	0.742	0.674	

image is determined as the average of the SSIM values from the three color bands, i.e. red, green and blue.

The objective of the study was twofold: Firstly to see how the proposed methods work and to identify the circumstances under which the different operations, orders and decisions work best. Secondly to compare the behavior of selected attributes when applied a compression operation (jpeg compression [42] at a quality of 75). The attributes that are compared are Area, Power, Volume, Vision, VisionP and entropy.

Implementation was in C programming language in the Cygwin platform and Matlab 6.5 in Windows.

A. Comparison of Decisions

Four restitution decisions discussed in Section II-1 were tested: the proposed NC, NN, and MP decisions, and the P_{mean} which is the best from those tested in [8]. The images were filtered by doing an area opening followed by a closing operation at an area threshold of $T = 150$. The results of filtering image Artificial using all four methods is demonstrated in Figure 1. It is clear that using P_{mean} visibly introduced different colors which is not the case with the new methods. This corresponds to the results in [8] that show visible quantization effects. Further scrutiny also shows that P_{mean} performs worse when the order used is based upon saturation or chromaticity.

The average results obtained from all the images are given in Table I and Table II. Table I shows that when filtered at the same threshold, the decision that produces the best quality is NC, closely followed by NN, MP and then P_{mean} . The average quality registered by using the nearest color is 0.777 which is equivalent to a 15% improvement when compared to using P_{mean} . Although the quality difference between NC and NN filters is not statistically significant, Table II shows that NC gives slightly better compression ratios. On the other hand, P_{mean} registers the highest compression ratios.

The overall results show that NC and NN orders produce high quality images devoid of colored artifacts.

B. Comparing the Preorders

In order to test the different types of preordering, all the 14 images were filtered using NN decision and at an area threshold of $T = 150$. It emerged that there is not a statistically significant difference between the performance of L_{HLS} and L_{lab} . This can be observed in Figure 3, 2 and Table I. These results show that L_{HLS} and L_{Lab} orders result in the best

TABLE II

COMPRESSION RATIOS OBTAINED AFTER FILTERING USING THE DIFFERENT DECISIONS

Image	NN	NC	MP	P_{mean}	AvCR
L_{Lab}	1.63	1.64	1.64	1.70	1.652
C	1.12	1.19	1.05	1.75	1.276
wL	1.64	1.64	1.64	1.65	1.643
wC	1.19	1.19	1.18	1.17	1.181
L_{HLS}	1.64	1.65	1.65	1.70	1.661
S	1.65	1.65	1.40	1.72	1.601
AvCR	1.478	1.491	1.423	1.616	

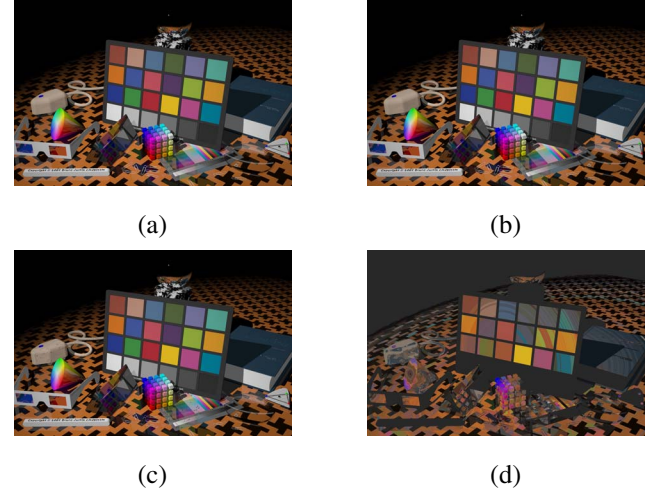


Fig. 1. Image Artificial after chromaticity filtering using (a) NC, (b) NN, (c) MP, and (d) P_{mean} , using area filter and area threshold 100

quality images and at the lowest sizes or best compression ratios. This therefore makes luminance based orders, on the average, better off than saturation or chromaticity ones.

However, scrutinizing individual images show that some are better off being filtered with chromaticity and saturation related orders. Figure 3 and 2 show a comparison of two images filtered at different thresholds ranging from 100 to 2000. Image Artificial is an image rich in colors and its analysis in Figure 2 shows that order chromaticity returns the best quality. On the other hand, image Leaves is mainly monochromatic and its analysis in Figure 3 shows that the luminance based orders register the best quality while chromaticity takes a back seat.

On average, the images that registered the worst quality are those by saturation, while chromaticity related filters register the lowest compression ratios as shown in Table II. Incidentally, the combination of luminance and chromaticity orders was always almost as good as using either of the two orders separately. This means that the proposed combination strategy needs further improvement. We therefore recommend that users choose either luminance, saturation or chromaticity orders until a more effective combination method is discovered.

C. Comparing the Attributes

The six attributes that were tested are: Area, Volume, Power, Vision, VisionP and Entropy. All the images were filtered at

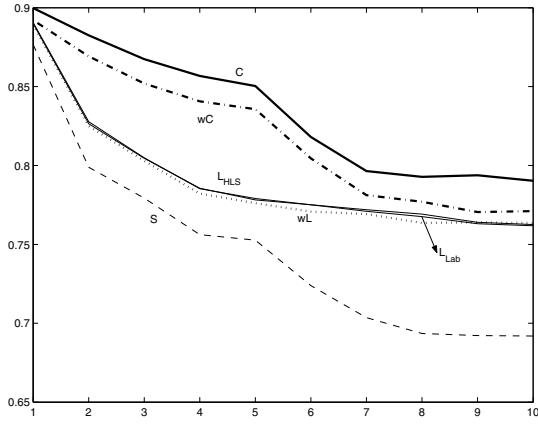


Fig. 2. Testing the Orders on image Artificial

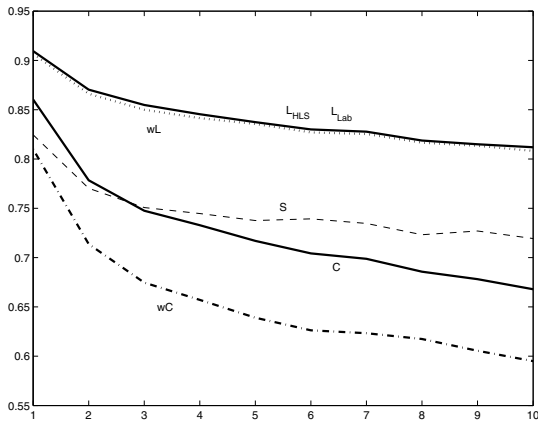


Fig. 3. Testing the Orders on image Leaves

10 different thresholds using the NC decision, L_{Lab} preorder and a MaxMin operation and then compressed. In order to obtain similar bit-rates (size in bytes * 8/ number of pixels), different threshold ranges for each of the attributes had to be obtained.

The experiments showed that all these attributes are suitable as a preprocessing filter for color compression although some attributes are better than others. For all images, quality reduces with an increase in compression or a reduction in bit rates. These results are shown in Figure 4 which illustrates how quality changes with a decrease in bit rate. It can be observed that the quality of Area, Volume and Power filters reduces gradually and predictably unlike Entropy, Vision and VisionP which at one point make drastic reductions which a small increase in filtering thresholds. The quality of Entropy filters sharply declines after an average bit rate of approximately 0.31 has been attained.

Vision and VisionP attributes give interesting results. An increase in filtering thresholds causes a reduction in quality until a turning point when it becomes fairly unpredictable. In some images, the quality begins to improve despite an increase in compression ratio. This happens at very low bit-rates and after the image has been severely degraded. This was also

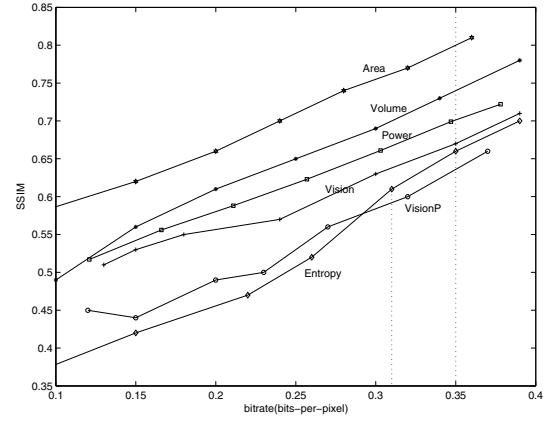


Fig. 4. Average quality expressed as SSIM over the 14 images in the data set as a function of bit-rate

shown to happen in [9] and attributed to their edge enhancing properties.

When Vision and VisionP attributes are compared, the quality of Vision filtered images is consistently better than VisionP filtered ones. This can be observed in Figure 4. This could mean that Volume a more robust attribute than Power.

In general, the attribute filter that results in the highest quality images is Area, followed by Volume, Power, Vision, and VisionP, while Entropy performs well before bit rates of 0.3.

IV. DISCUSSION AND CONCLUSIONS

This work proposes a color filtering method that is based on attribute filtering using the Max-tree representation. The major contribution of this research is in the proposition of a new method to convert the color image into a grayscale image that is then subjected to the conventional filtering techniques. Three ways of doing this are proposed: using saturation, chromacity or luminance components of the color image. We have also demonstrated three new and viable restitution mechanisms that reconstruct the filtered image back into color. The methods that have been discussed result in high quality images especially since no new colors or artifacts are introduced. The quality of the images generated using the best method have resulted in a quality improvement of 15% in comparison with a previous most similar method.

This work has also tested the above concepts on compression application. The image is filtered based on given attributes as a pre-processing step for compression. Six attributes were tested which are Area, Volume, Vision, VisionP, Entropy and Power. All these attributes were found to remove pscho-visually redundant information from an image. The best attributes that resulted in the highest quality images and the best compression ratios were Area and Volume. This performance is similar to earlier work [9] with the differences being attributed to a change in quality measurement metrics. The results have also shown that generally, luminance based images give better quality and lower sized images in image

compression. However, depending on the nature of the input image, chromaticity can play a better role too.

Future work can investigate better ways of combining the luminance, chromaticity and saturation values for generation of the ORDER image. Extensions to other connectivity, trees and vector-attribute filtering can also be explored.

April 20, 2012

REFERENCES

- [1] J. Serra, "Mathematical morphology for Boolean lattices," in *Image Analysis and Mathematical Morphology, II: Theoretical Advances*, J. Serra, Ed. London: Academic Press, 1988, ch. 2, pp. 37–58.
- [2] P. Salembier, A. Oliveras, and L. Garrido, "Anti-extensive connected operators for image and sequence processing," *IEEE Trans. Image Proc.*, vol. 7, no. 4, pp. 555–570, 1998.
- [3] P. Salembier and J. Serra, "Flat zones filtering, connected operators and filters by reconstruction," *IEEE Trans. Image Proc.*, vol. 4, no. 8, pp. 1153–1160, 1995.
- [4] H. J. A. M. Heijmans, "Connected morphological operators for binary images," *Comp. Vis. Image Understand.*, vol. 73, pp. 99–120, 1999.
- [5] E. J. Breen and R. Jones, "Attribute openings, thinnings and granulometries," *Computer Vision Image Understanding*, vol. 64, no. 3, pp. 377–389, 1996.
- [6] J. Angulo, "Morphological colour operators in totally ordered lattices based on distances: Application to image filtering, enhancement and analysis," *Computer Vision and Image Understanding*, vol. 107, pp. 56–73, 2007.
- [7] D. Gimenez and A. N. Evans, "An evaluation of area morphology scale-spaces for colour images," *Comp. Vis. Image Understand.*, vol. 110, pp. 32–42, 2008.
- [8] B. Naegel and N. Passat, "Component-trees and multi-value images: A comparative study," *LNCVS*, vol. 5720, 2009.
- [9] F. Tushabe and M. H. F. Wilkinson, "Image preprocessing for compression: Attribute filtering," in *Proceedings of International Conference on Signal Processing and Imaging Engineering (ICSPIE'07)*, San Francisco, USA, October 2007, pp. 1411–1418.
- [10] N. Young and A. N. Evans, "Psychovisually tuned attribute operators for pre-processing digital video," *IEE Proc. Vision, Image and Signal Processing*, vol. 150, pp. 277–286, 2003.
- [11] G. K. Ouzounis and M. H. F. Wilkinson, "Filament enhancement by non-linear volumetric filtering using clustering-based connectivity," in *Int. Workshop Intell. Comput. Pattern Anal. Synth. (IWICPAS) 2006*, ser. Lecture Notes in Computer Science, N. Zheng, X. Jiang, and X. Lan, Eds., vol. 4153, Xi'an, China, Aug 26–27 2006, pp. 317–327.
- [12] I. K. E. Purnama, K. Y. E. Aryanto, and M. H. F. Wilkinson, "Non-compactness attribute filtering to extract retinal blood vessels in fundus images," *International Journal of E-Health and Medical Communications*, 2010, in press.
- [13] M. H. F. Wilkinson and M. A. Westenberg, "Shape preserving filament enhancement filtering," in *Proc. MICCAI'2001*, ser. Lecture Notes in Computer Science, W. J. Niessen and M. A. Viergever, Eds., vol. 2208, 2001, pp. 770–777.
- [14] E. R. Urbach, J. B. T. M. Roerdink, and M. H. F. Wilkinson, "Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images," *tpami*, vol. 29, no. 2, pp. 272–285, 2007.
- [15] P. Salembier and M. H. F. Wilkinson, "Connected operators: A review of region-based morphological image processing techniques," *IEEE Signal Processing Magazine*, vol. 26, no. 6, pp. 136–157, 2009.
- [16] M. H. F. Wilkinson and G. K. Ouzounis, "Advances in connectivity and connected attribute filters," in *Advances in Imaging and Electron Physics*, P. W. Hawkes, Ed. Elsevier, 2010, vol. 161, pp. 211 – 275.
- [17] R. Jones, "Connected filtering and segmentation using component trees," *Comp. Vis. Image Understand.*, vol. 75, pp. 215–228, 1999.
- [18] L. Najman and M. Couprie, "Building the component tree in quasi-linear time," *IEEE Trans. Image Proc.*, vol. 15, pp. 3531–3539, 2006.
- [19] C. Ronse, "Why mathematical morphology needs complete lattices," *Signal Processing*, vol. 21, no. 2, pp. 129 – 154, 1990.
- [20] J. Serra, "Connectivity on complete lattices," *J. Math. Imag. Vis.*, vol. 9, no. 3, pp. 231–251, 1998.
- [21] O. Lezoray, C. Meurie, and A. Elmoataz, "A graph approach to color mathematical morphology," in *Proceedings of the 5th IEEE International Symposium on Signal Processing and Information Technology*, 2005, pp. 856–861.
- [22] K. N. Plataniotis and A. N. Venetsanopoulos, *Color image processing and applications*. New York, NY, USA: Springer-Verlag New York, Inc., 2000.
- [23] V. Witte, S. Schulte, M. Nachtegaal, D. Weken, and E. E. Kerre, "Vector morphological operators for colour images," *LNCVS*, vol. 3656/2005, pp. 667–675, 2005.
- [24] M. Wheeler and M. A. Zmuda, "Processing color and complex data using mathematical morphology," in *Proc. IEEE National Aerospace and Electronics Conference (NAECON) 2000*, 2000, pp. 618 –624.
- [25] A. Meijster and M. H. F. Wilkinson, "A comparison of algorithms for connected set openings and closings," *tpami*, vol. 34, no. 4, pp. 484–494, 2002.
- [26] L. Vincent, "Greyscale area openings and closings, their efficient implementation and applications," in *Proceedings EURASIP Workshop on Mathematical Morphology and its Application to Signal Processing*, 1993, pp. 22–27.
- [27] R. E. Tarjan, "Efficiency of a good but not linear set union algorithm," *ACM*, vol. 22, pp. 215–225, 1975.
- [28] G. K. Ouzounis and M. H. F. Wilkinson, "Hyperconnected attribute filters based on k -flat zones," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010, in press.
- [29] E. R. Urbach and M. H. F. Wilkinson, "Shape-only granulometries and grey-scale shape filters," in *Proc. Int. Symp. Math. Morphology (ISMM) 2002*, Sydney, Australia, April 2002, pp. 305 – 314.
- [30] A. N. Evans and D. Gimenez, "Extending connected operators to colour images," in *Proceedings of the IEEE International Conference on Image Processing, 2008*, San Diego, California, U.S.A., October 2008, pp. 2184–2187.
- [31] A. N. Evans, "Vector area morphology for motion field smoothing and interpretation," *IEE Proc.-Vis. Image Signal Process.*, vol. 150, pp. 219–226, 2003.
- [32] S. Gibson, R. Harvey, and G. D. Finlayson, "Convex colour sieves," *Lecture Notes in Computer Science*, vol. 2695/2003, p. 1079, 2004.
- [33] P. Salembier and L. Garrido, "Binary partition tree as an efficient representation for image processing, segmentation and information retrieval," *IEEE Trans. Image Proc.*, vol. 9, no. 4, pp. 561–576, April, 2000.
- [34] V. Vilaplana, F. Marques, and P. Salembier, "Binary partition trees for object detection," *IEEE Trans. Image Proc.*, vol. 17, no. 11, pp. 1–16, November 2008.
- [35] HunterLab, "Cie l*a*b* color scale," vol. 8, no. 7, 2008.
- [36] F. Cheng and A. N. Venetsanopoulos, "An adaptive morphological filter for image processing," *IEEE Trans. Image Proc.*, vol. 1, no. 4, pp. 533–539, 1992.
- [37] C. Vachier, "Utilisation d'un critère volumique pour le filtrage des images," in *Proc. Reconnaissance des Formes et Intelligence Artificielle*, vol. 1, Clermont-Ferrand, 1998, pp. 307–315.
- [38] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379 – 423, 1948.
- [39] M. Dimiccoli and P. Salembier, "Perceptual filtering with connected operators and image inpainting," in *Proc. Int. Symp. Math. Morphology (ISMM) 2007*, G. J. F. Banon, J. Barrera, U. d. M. Braga-Neto, and N. S. T. Hirata, Eds. Universidade de São Paulo (USP), 2007, pp. 227–238.
- [40] S. Garg, "Image compression benchmark," 2008.
- [41] Z. Wang, A. C. Bovik, and H. R. Sheikh, "Structural similarity based image quality assessment," in *Digital Video Image Quality and Perceptual Coding*, ser. Series in Signal Processing and Communications, H. R. Wu and K. R. Rao, Eds. CRC, 2005, pp. 225–241.
- [42] JPEG, "Information technology-coded representation of picture and audio information-digital compression and coding of continuous-tone still images (jpeg standard)," ISO and ITU-T, Computing Science Technical Report ITU-T Rec. T.81—ISO/IEC 10918-1, 1993.